UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/898,708	07/03/2001	James J. Babka	027-0006	2440

22120 7590 03/23/2004

ZAGORIN O'BRIEN & GRAHAM, L.L.P.
7600B N. CAPITAL OF TEXAS HWY.
SUITE 350
AUSTIN, TX 78731

EXAMINER	
TRUONG, CAM Y T	
ART UNIT	**PAPER NUMBER**
2172	5

DATE MAILED: 03/23/2004

Please find below and/or attached an Office communication concerning this application or proceeding.

4

Office Action Summary	Application No.	Applicant(s)	
	09/898,708	BABKA ET AL.	
	Examin r	Art Unit	
	Cam Y T Truong	2172	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *09 January 2004*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-41* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-41* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☐ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: _____.

DETAILED ACTION

1. Applicant's arguments filed 1/9/04 have been fully considered but they are not persuasive.

Applicant argued that Jeyaraman does not teach collapsing in an order-insensitive manner. Jeyaraman teaches the system starts with an old tree and a new tree. The system first matches leaf nodes of old tree and new tree. The system may look for exact matches or partial matches of the data stored in the leaf nodes. The system generates a node split operation for the parent. The parent node is split into a first parent node and a second parent node. The first parent node inherits all of the children that are present in new_t and the second parent inherits the remaining children. CLP is an operation collapses the contents of a first node and a second node. The resulting node gets the same tag type as the first node. The children of the second node become the right-most children of the resulting node. Since this tree is an unsorted tree; thus, the collapsing the contents of a first node and a second node is in an order-insensitive manner (col. 9, lines 20-25; col. 7, lines 60-64).

Applicant argued that Jeyaraman does not teach "recursively collapsing sub-hierarchies thereof using encodings that, at least at a same level thereof; includes orthogonal values". Jeyaraman teaches that if a parent node in old_t does not have all of the same children in new_t, the system generates a node split operation for the parent, splitting the parent node into a first parent and a second parent at step 510. The first parent inherits all of the children that are present in new_t, and the second parent inherits the remaining children. It a parent node in old_t has all of the same children

and additional children in new_t, the system generates a node collapse operation to bring all the children together in new_t at step 512. Additionally, if all of the children of a first parent in old_t move to a second parent in new_t, the system generates a node collapse operation to collapse the first parent into the second parent so that all of the children of the first parent are inherited by the second parent. The system repeats these steps for ascending levels of the tree. The above information shows that the system recursively collapses sub-tree using new_ts. Parent children in old_t is represented as orthogonal values (col. 7, lines 50-66; col. 8, lines 1-10).

For the above reason, examiner believed that rejection of the last office action was proper.

Claim Rejections - 35 USC § 103

2. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

3. Claims 1-4, 12, 14, 15, 17, 18, 19, 24, 26-32, 35-38, 40-41 are rejected under 35 U.S.C. 103(a) as being unpatentable over Jeyaraman (USP 6311187).

As to claim 1, Jeyaraman teaches the claimed limitations:

"collapsing plural nodes thereof into respective representations that each incorporate information of a respective node and that of any child nodes thereof"

as if all of the children of a first parent in old_t move to a second parent in new_t, the

system generates a node collapse operation to collapse the first parent into the second parent so that all of the children of the first parent are inherited by the second parent. The system repeats these steps for ascending levels of the tree (col. 7, lines 50-66; col. 8, lines 1-10);

"and based on correspondence of particular instances of the collapsed representations, identifying the respective portions as equivalent" as matching leaf nodes T1 and T2. If two leaf nodes have the same content, then the hash function generates the same identifier. Where the leaf nodes actually contain data and value identifiers. This information shows that the system maps the contents of nodes with each other. The contents of nodes are data and value identifiers. Each data and value identifier is presented as a power-of-two encoded (col. 9, lines 35-46; col. 8, lines 20-25).

Jeyaraman does not clearly teach the claimed limitation "wherein the collapsing is, order-insensitive with respect to information of the respective child nodes". However, Jeyaraman teaches the system starts with an old tree and a new tree. The system first matches leaf nodes of old tree and new tree. The system may look for exact matches or partial matches of the data stored in the leaf nodes. The system generates a node split operation for the parent. The parent node is split into a first parent node and a second parent node. The first parent node inherits all of the children that are present in new_t and the second parent inherits the remaining children. CLP is an operation collapses the contents of a first node and a second node. The resulting node gets the same tag type as the first node. The children of the second node become the right-

most children of the resulting node. Since this tree is an unsorted tree, thus, the collapsing the contents of a first node and a second node is in an order-insensitive manner (col. 9, lines 20-25; col. 7, lines 60-64).

It would have been obvious to a person of an ordinary skill in the art at the time the invention was made to apply Jeyaraman's teaching of splitting the parent node into a first parent node and a second parent node and collapsing the contents of a first node and a second node in order to eliminate duplicate nodes in a tree for saving memory space.

As to claim 2, Jeyaraman teaches the claimed limitation " wherein the collapsed representations include respective aggregations of orthogonally-encoded child node information" as a tree may represent a document consisting of sections, paragraphs and individual sentences containing parsable character data. All nodes in fig. 6A include a name tag, a value, and an associated value identifier. The parent node is split into a first parent node and a second parent node. The first parent node inherits all of the children that are present in new_t and the second parent inherits the remaining children. This information shows that this document is encoded into nodes of tree. Thus, when a parent node is collapsed, the first and second parent nodes include respective aggregations of encoded node children. The first and second parent nodes are presented as the collapsed representation (col. 8, lines 15-25; col. 7, lines 59-65).

As to claim 3, Jeyaraman teaches the claimed limitation "wherein a unit of orthogonally- encoded child node information includes a power-of-two encoded mapping of a concatenation of the child node information with a similarly encoded mapping of respective information of child nodes thereof". However, Jeyaraman teaches matching leaf nodes T1 and T2. If two leaf nodes have the same content, then the hash function generates the same identifier. Where the leaf nodes actually contain data and value identifiers. This information shows that the system maps the contents of nodes with each other. The contents of nodes are data and value identifiers. Each data and value identifier is presented as a power-of-two encoded (col. 9, lines 35-46; col. 8, lines 20-25).

As to claim 4, Jeyaraman teaches the claimed limitation

"wherein a unit of orthogonally- encoded child node information includes a

power-of-two encoded mapping of a concatenation of the child node information with recursively encoded mappings of respective subhierarchies thereof" as mapping leaf nodes tree 1 and tree 2. For each level_l in T2 (leaf to the root) { to_be_completed_list = list of all the node value identifiers at level_l in T2. If the node in the to_be_completed_list is the root node, find the matching node t in T1. Where the leaf nodes actually contain data and value identifiers. The modification phase brings together the children of internal nodes, in a bottom-up fashion. This involves scanning all the nodes from the bottom-most level and scanning each level until level zero is reached. Note that the identity of each internal node is established by the collective

identify of its children. The above information shows the system maps the contents of nodes in T1 and T2 recursively by following a bottom-up fashion. Data and value identifier of a node in a tree is presented as power-of-two encoded value (col. 11, lines 15-43; col. 9, lines 35-46; col. 8, lines 20-25).

As to claim 12, Jeyaraman teaches the claimed limitation "wherein the corresponding collapsed representations is based on identity of respective mapped codes" as mapping leaf nodes tree 1 and tree 2. For each level_l in T2 (leaf to the root) { to_be_completed_list = list of all the node value identifiers at level_l in T2. If the node in the to_be_completed_list is the root node, find the matching node t in T1. Where the leaf nodes actually contain data and value identifiers (col. 11, lines 15-43; col. 9, lines 35-46; col. 8, lines 20-25).

As to claim 14, Jeyaraman teaches the claimed limitations:

"wherein the hierarchically-organized data structure includes at least three levels of nodes" as (fig. 6D-6F);

"and further comprising performing the collapsing at successive ones of the levels of the hierarchically-organized data structure" as (fig. 6D-6F, col. 7, lines 57-67).

As to claim 15, Jeyaraman teaches the claimed limitation "wherein the hierarchically-organized data structure includes a tree-organized data structure" as (fig. 6D)

As to claim 17, Jeyaraman teaches the claimed limitations "wherein the hierarchically-organized data structure encodes subassembly information as sub-Hierarchies thereof and encodes component parts at least at leaf nodes thereof" as (col. 8, lines 15-29).

As to claim 18, Jeyaraman teaches the claimed limitations:

"associating a first-level identifier with each of plural leaf nodes at a first-level of the tree" as all nodes in fig. 6A include a name, a value, and an associated value identifier. Node D is root node in the first level of the tree. Thus, node D identifier is a first-level identifier with each of plural leaf nodes below (col. 8, lines 30-35), "wherein distinct leaf node values are associated with distinct first identifiers" as generating a unique identifier for each of the leaf nodes in T2 based on the content of the leaf node. This can be accomplished by using a hash function to generate a unique identifier for each of the leaf nodes. If two leaf nodes have the same content, then the hash function generates the same identifier. This information means that if two leaf nodes have different value, the system assigns different identifiers to them. Thus, different leaf node content are associated with different identifiers (col. 9, lines 40-48) "and equivalent leaf node values are associated with same first identifiers" as (col. 9, lines 40-48);

"each next level of the tree, associating an identifier with each node thereof" as (fig. 6A, col. 8, lines 30-35), "each such identifier including a current node contribution and a contribution associated with any child nodes thereof" as all nodes in fig. 6A

include a name, a value, and an associated value identifier. The parent node Se on the left side of the tree has an identifier. This parent node includes a current node contribution such as node P on the left side of the tree. This node P associated with child nodes. Thus, the identifier of the left Se has to include P node and all children nodes of P. Similarly, the identifier of the right Se has to include mild P node and all children nodes of mild P (fig. 6A, col. 8, lines 30-35), "wherein the child nodes contribution is computed using a combining function operative on identifiers associated with the child nodes" as the system generates a node collapse operation to bring all the children together in new_t (new tree). Since all nodes include a name, a value, an associated value identifier, thus, when bringing all nodes together, the system has to bring all nodes identifiers too (col. 65-67; col. 8, lines 20-25);

"and wherein for a second level of the tree, respective child nodes are the leaf nodes of the first-level of the tree" as (fig. 6A).

Jeyaraman does not clearly teach the claimed limitation "wherein the identifiers and combining function are selected to ensure that same combinations of child node identifiers result in same child nodes contributions irrespective of ordering of the child node identifiers". However, Jeyaraman teaches that the system first matches leaf nodes of old_t (old tree) and new_t (new tree). If a parent node in old_t has all of the same children and additional children in new_T, the system generates a node collapse operation to bring all the children together in new_t. All nodes include value identifiers. The above information shows that when system matches nodes from trees, the system has to match their identifiers too. Thus, when system brings all the matched children

together from old_tree to new_t, the system ensures that same combinations of child node identifiers result in same child nodes in new_t without ordering of the child node identifiers (col. 7, lines 40-65; col. 8, lines 15-25).

It would have been obvious to a person of an ordinary skill in the art at the time the invention was made to apply Jeyaraman's teaching of bringing together old_t matched nodes with new_t nodes to new_t and all nodes contain value identifiers in order to save memory space, eliminate redundant node, and to transform a old tree to new tree efficiently.

As to claim 19, Jeyaraman teaches the claimed limitation "wherein the identifiers are orthogonally-encoded mappings of respective string encodings of the current node contribution concatenated with respective orthogonally-encoded mappings of child node information" as (col. 11, lines 15-45).

As to claim 24, Jeyaraman teaches the claimed limitation "at least at any particular level of the tree-oriented data representation, the identifiers are orthogonally-encoded" as (col. 11, lines 35-45).

As to claim 26, Jeyaraman teaches the claimed limitation "employed in a duplicate elimination operation on the tree-oriented data representation" as (col. 9, lines 20-25).

As to claim 27, Jeyaraman teaches the claimed limitation "employed in a duplicate identification operation on the tree-oriented data representation" as (col. 9, lines 40-50).

As to claim 28, Jeyaraman teaches the claimed limitation "employed in an equality test operation on portions of the tree-oriented data representation" as (col. 11, lines 35-47).

As to claim 29, Jeyaraman teaches the claimed limitations:

"representing any given node of the hierarchically-organized data as a concatenation of node-specific information with a combination of the orthogonal values for each collapsed sub-hierarchy therebeneath" as (col. 7, lines 50-66; col. 8, lines 1-10).

Jeyaraman does not clearly teach the claimed limitation "recursively collapsing sub-hierarchies thereof using encodings that, at least at a same level thereof; includes orthogonal values" as if a parent node in old_t does not have all of the same children in new_t, the system generates a node split operation for the parent, splitting the parent node into a first parent and a second parent at step 510. The first parent inherits all of the children that are present in new_t, and the second parent inherits the remaining children. It a parent node in old_t has all of the same children and additional children in new_t, the system generates a node collapse operation to bring all the children together in new_t at step 512. Additionally, if all of the children of a first parent in old_t move to a

second parent in new_t, the system generates a node collapse operation to collapse the first parent into the second parent so that all of the children of the first parent are inherited by the second parent. The system repeats these steps for ascending levels of the tree. The above information shows that the system recursively collapses sub-tree using new_ts. Parent children in old_t is represented as orthogonal values (col. 7, lines 50-66; col. 8, lines 1-10).

It would have been obvious to a person of an ordinary skill in the art at the time the invention was made to apply Jeyaraman's teaching of repeating of generating a node collapse operation to bring all the children from old_t to new_t and to collapse the first parent into the second parent so that all of the children of the first parent are inherited by the second parent in order to maintain tree structure and save memory space.

As to claim 30, Jeyaraman teaches the claimed limitation "transforming from a first encoding of the hierarchically-organized data to a collapsed second form" as (col. 7, lines 65-67; col. 8, lines 1-10).

As to claim 31, Jeyaraman teaches the claimed limitation "employed to eliminate duplicate sub-hierarchies in the hierarchically organized data" as (figs.6A-6B).

As to claim 32, Jeyaraman teaches the claimed limitation "employed to collapse duplicate sub-hierarchies in the hierarchically-organized data, wherein the

concatenation further includes a count of duplicate sub-hierarchies collapsed beneath any given node" as (col. 7, lines 40-65; col. 12, lines 25-35).

As to claim 35, Jeyaraman teaches the claimed limitations:

"an encoding of a hierarchically-organized data structure instantiable in memory addressable by the one or more processors" as (col. 6, lines 60-67; fig. 1);

"instructions executable by the one or more processors to operate on at least one instance of the hierarchically-organized data structure instantiated in memory, the instructions" as (fig. 1, col. 5, lines 55-67), "when executed, causing the processor to define a counterpart data structure in the memory by collapsing plural nodes of the hierarchically-organized data structure into respective representations that each incorporate information of a respective node and that of any child. nodes thereof" (fig. 1, col. 5, lines 55-67, col. 7, lines 25-65).

Jeyaraman does not clearly teach the claimed limitation "wherein the collapsing includes an order-insensitive aggregation of orthogonal encodings of information of the respective child nodes." However, Jeyaraman teaches that the system generates a node split operation for the parent. The parent node is split into a first parent node and a second parent node. The first parent node inherists all of the children that are present in new_t and the second parent inherits the remaining children. CLP is an operation collapses the contents of a first node and a second node. The resulting node gets the same tag type as the first node. The children of the second node become the right-most children of the resulting node. This information implies that splitting a node is

order-insensitive with respect to information of the respective child nodes. The content of the first parent node and the second parent node is represented as information of the respective child nodes (col. 9, lines 20-25; col. 7, lines 60-64).

It would have been obvious to a person of an ordinary skill in the art at the time the invention was made to apply Jeyaraman's teaching of splitting the parent node into a first parent node and a second parent node and collapsing the contents of a first node and a second node in order to eliminate duplicate nodes in a tree for saving memory space.

As to claim 36, Jeyaraman teaches the claimed limitation "matching instructions executable by the one or more processors to identify distinct sub-hierarchies of the hierarchically-organized data structure as at least equivalent based on correspondence of the collapsed representations" as (col. 7, lines 40-65; col. 9, lines 15-65; col. 4, lines 20-45).

As to claim 37, Jeyaraman teaches the claimed limitation "matching instructions executable by the one or more processors to identify at least equivalent portions of first and second ones of the hierarchically organized data structure based on correspondence of collapsed representations thereof" as (col. 9, lines 15-65; col. 4, lines 20-45).

As to claim 38, Jeyaraman teaches the claimed limitation "wherein the order insensitive aggregation is performed recursively at successive levels of a collapsed sub-hierarchy" as (col. 11, lines 15-43; col. 9, lines 35-46; col. 8, lines 20-25).

As to claim 40, Jeyaraman teaches the claimed limitations:

"wherein the hierarchically-organized data structure encodes and a subassembly decomposition of a product configuration" as (figs. 6A-6C; col. 8, lines 15-30);

" and wherein the information management tool further identifies, based on correspondence of collapsed representations of the hierarchically organized data structure, equivalent sub-assemblies without regard to ordering of elements thereof" as (col. 7, lines 25-65).

As to claim 41, Jeyaraman teaches the claimed limitations:

"a processor and memory addressable thereby" as (col. 4, lines 37-45).

Jeyaraman does not clearly teach the claimed limitation "and means for performing an element order independent comparison of hierarchically organized data structures using a transformation operation that orthogonally and recursively encodes child node information". However, Jeyaraman teaches that as mapping leaf nodes tree 1 and tree 2. For each level_I in T2 (leaf to the root) { to_be_completed_list = list of all the node value identifiers at level_I in T2. If the node in the to_be_completed_list is the root node, find the matching node t in T1. Where the leaf nodes actually contain data

and value identifiers. The modification phase brings together the children of internal nodes, in a bottom-up fashion. This involves scanning all the nodes from the bottom-most level and scanning each level until level zero is reached. Note that the identity of each internal node is established by the collective identify of its children. The system transforms old_tree into new_tree. The above information shows the system maps the contents of nodes in T1 and T2 recursively by following a bottom-up fashion (col. 11, lines 15-43; col. 9, lines 35-46; col. 8, lines 5-25).

It would have been obvious to a person of an ordinary skill in the art at the time the invention was made to apply Jeyaraman's teaching of scanning all the nodes from bottom-most level to compare leaf nodes in trees for transforms old_t into new_t in order to update a tree structure.

4. Claim 5 is rejected under 35 U.S.C. 103(a) as being unpatentable over Jeyaraman in view of Aggarwal et al (or "hereinafter" Aggarwal") (USP 5781906).

As to claim 5, Jeyaraman discloses the claimed limitation subject matter in claim 1, except the claimed limitation "wherein the order-insensitive collapsing includes an arithmetic sum of orthogonal binary encodings of child node information". However, Aggarwal teaches in step 201, a binary tree is constructed such that the entries in the leaf nodes correspond to multidimensional also called spatial data objects stored on DASD 105. Thus, all non-leaf nodes have branch factors of 2. Let A be an internal node of the binary tree with children B.sub.1 and B.sub.2. A skew factor near 1/2 implies that the tree will be quite well

balanced, but may not do as well with respect to some or all of the design objectives. The leaf number N.sub.A of a node A is defined as the total number of data objects in the leaf descendants of that node. The area of an aligned rectangle I is denoted by A(I). The method ensures that the leaf numbers N.sub.B.sbsb.1 and N.sub.B.sbsb.2 of each of these children are each at least p.multidot.N.sub.A. Among all partitions examined which satisfy this leaf number condition, the one chosen minimizes the sum of the areas of the minimum bounding rectangles A(I.sub.B.sbsb.I)+A(I.sub.B.sbsb.I) subject to a predetermined overlap factor constraint (col. 5, lines 50-67).

It would have been obvious to a person of an ordinary skill in the art at the time the invention was made to apply Aggrawal's teaching of in a binary tree, the leaf numbers N.sub.B.sbsb.1 and N.sub.B.sbsb.2 of each of these children are each at least p.multidot.N.sub.A. Among all partitions examined which satisfy this leaf number condition, the one chosen minimizes the sum of the areas of the minimum bounding rectangles A(I.sub.B.sbsb.I)+A(I.sub.B.sbsb.I) subject to Jeyaraman's system in order to save memory space.

5. Claims 6, 7, 13, 20-23, 25, 33, 34, 39 are rejected under 35 U.S.C. 103(a) as being unpatentable over Jeyaraman in view of Brown (USP 6539369).

As to claim 6, Jeyaraman discloses the claimed limitation subject matter in claim 1, except the claimed limitation "wherein distinct tables are defined for each level of the hierarchically organized data structure". However, Brown teaches that lookup table 100

includes mapper level_1 112a and 112b. Mapper level_1 112a includes the first 16 of 32 levels of the binary tree. Mapper level_2 112b includes the next 8 levels of the 32-level binary tree (col. 5, lines 30-65).

It would have been obvious to a person of an ordinary skill in the art at the time the invention was made to apply Brown's teaching of mapping tree levels to lookup table levels to Jeyaraman's system in order to allow multiple sparse subtree descriptors to be stored in a subtree entry in a memory.

As to claim 7, Jeyaraman discloses the claimed limitation subject matter in claim 1, except the claimed limitation "wherein a table spans multiple levels of the hierarchically-organized data structure". However, Brown teaches that a lookup table, which allows sparse, subtree descriptiors and dense subtree descriptors to be stored in the same memory (abstract). It would have been obvious to a person of an ordinary skill in the art at the time the invention was made to apply Brown's teaching of a lookup table which allows sparse subtree descriptiors and dense subtree descriptors Jeyaraman's system in order to store nodes of tree in same memory.

As to claim 13, Jeyaraman discloses the claimed limitation subject matter in claim 1, except the claimed limitation "wherein the order-insensitive collapsing includes an arithmetic addition of orthogonally-encoded values that index into a store of child node information". However, Brown teaches that search for a pointer requires the following cache memory accesses: (1) read a 16 bit code word 46; (2) read a 16-bit

base address 42; (3) read a 4 bit offset 54 from the map table 32; (4) read a pointer at a pointer index where the pointer index is the sum of the base address 42, the code word offset 46a and the 4-bit offset 54.

It would have been obvious to a person of an ordinary skill in the art at the time the invention was made to apply Brown's teaching of the pointer index is the sum of the base address 42, the code word offset 46a and the 4-bit offset 54 to Jeyaraman's system in order to allow search/retrieve data without need for a sequential search through the collection of elements.

As to claim 20, Jeyaraman discloses the claimed limitation in claim 18, except he claimed limitation "wherein the orthogonally-encoded mappings at each level of the tree-oriented data representation are in accordance with a corresponding level-specific table". However, Brown teaches that lookup table 100 includes mapper level_1 112a and 112b. Mapper level_1 112a includes the first 16 of 32 levels of the binary tree. Mapper level_2 112b includes the next 8 levels of the 32-level binary tree (col. 5, lines 30-65).

It would have been obvious to a person of an ordinary skill in the art at the time the invention was made to apply Brown's teaching of mapping tree levels to lookup table levels to Jeyaraman's system in order to allow multiple sparse subtree descriptors to be stored in a subtree entry in a memory.

As to claim 21, Jeyaraman discloses the claimed limitation in claim 18, except the claimed limitation "wherein the orthogonally-encoded mappings for distinct portions of the tree oriented data representation are in accordance with respective tables". However, Brown teaches mapping node 130^23 in master lookup table 200a and mapping node 130^4 in slave lookup table 200b (col. 14, lines 25-50).

It would have been obvious to a person of an ordinary skill in the art at the time the invention was made to apply Brown's teaching of mapping node 130^23 in master lookup table 200a and mapping node 130^4 in slave lookup table 200b to Jeyaraman's system in order to search/retrieve deep levels of a tree.

As to claim 22, Jeyaraman discloses the claimed limitation in claim 18, except the claimed limitation "wherein the orthogonally-encoded mappings for multiple levels of the tree oriented data representation are in accordance with a single corresponding hash table". However, Brown teaches that lookup table 100 includes mapper level_1 112a and 112b. Mapper level_1 112a includes the first 16 of 32 levels of the binary tree. Mapper level_2 112b includes the next 8 levels of the 32-level binary tree (col. 5, lines 30-65).

It would have been obvious to a person of an ordinary skill in the art at the time the invention was made to apply Brown's teaching of mapping tree levels to lookup table levels to Jeyaraman's system in order to allow multiple sparse subtree descriptors to be stored in a subtree entry in a memory.

As to claim 23, Jeyaraman discloses the claimed limitation in claim 18, except the claimed limitation "wherein the orthogonally-encoded hashes for each level of the tree-oriented data representation are in accordance with a single corresponding table". However, Brown teaches that lookup table 100 includes mapper level_1 112a and 112b. Mapper level_1 112a includes the first 16 of 32 levels of the binary tree. Mapper level_2 112b includes the next 8 levels of the 32-level binary tree . Each nodes includes string bits (col. 5, lines 30-65; col. 9, lines 25-60).

It would have been obvious to a person of an ordinary skill in the art at the time the invention was made to apply Brown's teaching of mapping tree levels to lookup table levels to Jeyaraman's system in order to allow multiple sparse subtree descriptors to be stored in a subtree entry in memory in order.

As to claim 25, Jeyaraman teaches the claimed limitation "wherein the combining function includes addition" as the system generates a node collapse operation to bring all the children together in new_t. This information implies that the system has included a combining function includes addition (col. 7, lines 65-66).

Jeyaraman fails to teach the claimed limitation "wherein the identifiers correspond to orthogonal binary encodings of integers". However, Brown teaches that the route index 102 for level-5 nodes 130^9-130^12 is r1, thus locations 140^9, 140^10 at addresses 01000 and 01001 in the L1 mapper 106a store r1 (col. 6, lines48-50).

It would have been obvious to a person of an ordinary skill in the skill in the art at the time the invention was made to apply Brown's teaching of that the route index 102

for level-5 nodes 130^9-130^12 is r1, thus locations 140^9, 140^10 at addresses 01000 and 01001 in the L1 mapper 106a store r1 to Jeyaraman's system in order to store nodes in memory efficiently.

As to claim 33, Jeyaraman teaches the claimed limitations:

"a program sequence including a recursively called set of instructions executable by one or more processors to operate on at least one instance of an hierarchically-organized data structure, the instructions" as if a parent node in old_t does not have all of the same children in new_t, the system generates a node split operation for the parent, splitting the parent node into a first parent and a second parent at step 510. The first parent inherits all of the children that are present in new_t, and the second parent inherits the remaining children. It a parent node in old_t has all of the same children and additional children in new_t, the system generates a node collapse operation to bring all the children together in new_t at step 512. Additionally, if all of the children of a first parent in old_t move to a second parent in new_t, the system generates a node collapse operation to collapse the first parent into the second parent so that all of the children of the first parent are inherited by the second parent. The system repeats these steps for ascending levels of the tree. The above information shows that the system recursively collapses sub-tree using new_ts. Parent children in old_t is represented as orthogonal values (col. 7, lines 50-66; col. 8, lines 1-10),

"when executed, causing the processor to define a counterpart data structure by collapsing plural nodes of the hierarchically-organized data structure into respective

representations that each incorporate information of a respective node and that of any child nodes thereof" as (fig. 1, col. 5, lines 55-67), "wherein the collapsing includes an order-insensitive aggregation of orthogonal encodings of information of the respective child nodes" as the system generates a node split operation for the parent. The parent node is split into a first parent node and a second parent node. The first parent node inherits all of the children that are present in new_t and the second parent inherits the remaining children. CLP is an operation collapses the contents of a first node and a second node. The resulting node gets the same tag type as the first node. The children of the second node become the right-most children of the resulting node. This information implies that splitting a node is order-insensitive with respect to information of the respective child nodes. The content of the first parent node and the second parent node is represented as information of the respective child nodes (col. 9, lines 20-25; col. 7, lines 60-64);

"wherein values thereof provide the orthogonal encodings and keys thereof combine the information of respective nodes with an aggregation of the collapsed representations for child nodes thereof" as (col. 8, lines 15-25; col. 7, lines 50-67).

Jeyaraman fails to teach the claimed limitation "and an object implementing the counterpart data structure including at least one table". However, Brown teaches a binary tree representation of the entries stored in the mappers 106a-c in the lookup table 100 (col. 5, lines 45-47).

It would have been obvious to a person of an ordinary skill in the art at the time the invention was made to apply Brown's teaching of a binary tree representation of the

entries stored in the mappers 106a-c in the lookup table 100 to Jeyaraman's system in order to reduce number search depth levels of a tree during matching nodes.

As to claim 34, Jeyaraman teaches the claimed limitation "wherein the at least one computer readable medium is selected from the set of a disk, tape or other magnetic, optical, or electronic storage medium and a network, wire line, wireless or other communications medium" as (col. 2, lines 60-67).

As to claim 39, Jeyaraman teaches the claimed limitations:

"a recursively encoded mapping wherein, for any particular node of the hierarchically-organized data structure" as mapping leaf nodes tree 1 and tree 2. For each level_l in T2 (leaf to the root) { to_be_completed_list = list of all the node value identifiers at level_l in T2. If the node in the to_be_completed_list is the root node, find the matching node t in T1. Where the leaf nodes actually contain data and value identifiers. The modification phase brings together the children of internal nodes, in a bottom-up fashion. This involves scaning all the nodes from the bottom-most level and scanning each level until level zero is reached. Note that the identity of each internal node is established by the collective identify of its children. The above information shows the system maps the contents of nodes in T1 and T2 recursively by following a bottom-up fashion. Data and value identifier of a node in a tree is presented as power-of-two encoded value (col. 11, lines 15-43; col. 9, lines 35-46; col. 8, lines 20-25).

" wherein, at least for same-level nodes of the hierarchically-organized data structure, corresponding values are orthogonal" as (fig. 6A-6B, col. 8, lines 15-30).

Jeyaraman fails to teach the claimed limitation "a corresponding table entry encodes both respective values for child nodes thereof in accordance with the order-insensitive information and aggregation associated with the particular node itself at least one hash table". However, Brown teaches a binary tree representation of the entries stored in the mappers 106a-c in the lookup table 100 (col. 5, lines 45-47).

It would have been obvious to a person of an ordinary skill in the art at the time the invention was made to apply Brown's teaching of a binary tree representation of the entries stored in the mappers 106a-c in the lookup table 100 to Jeyaraman's system in order to reduce number search depth levels of a tree during matching nodes.

Allowable Subject Matter

6. Claims 8-11 and 16 are objected to as being dependent upon a rejected base claim, but would be allowable if rewritten in independent form including all of the limitations of the base claim and any intervening claims.

As to claim 8, none of the available prior art of record teaches or fairly suggest an arithmetic addition of orthogonal binary encodings that identify corresponding table entries for respective child nodes; and concatenation of a result of the arithmetic addition with an encoding of information for the particular node". Combining contents of nodes is well known in the art as taught by Jeyaraman. However, prior art such as

Jeyaraman does not teach "an arithmetic addition of orthogonal binary encodings.....particular node" in the specific combination as recited in claim 8.

Conclusion

7. **THIS ACTION IS MADE FINAL.** Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

Contact Information

8. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Cam-Y Truong whose telephone number is (703 -605-1169). The examiner can normally be reached on Mon - Fri from 8:00AM to 4:00PM. If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Kim Vu, can be reached on (703-305-4393). The fax phone number for organization where this application or proceeding is assigned is (703-308-9051). Any inquiry of a general nature or relating to the status of this application or proceeding should be directed to the receptionist whose telephone number is (703-305-3900).

Cam-Y Truong

9/26/03

SHAHID ALAM
PRIMARY EXAMINER